**주 간 회 의 록**

|  |  |  |  |
| --- | --- | --- | --- |
| **팀 명** | Sky Rescuer | **차수** | 6 차 |
| **일 시** | 2023년 5월 14일 일 요일 19시 00분 | | |
| **장 소** | 창업보육센터 C동 102호 | | |
| **참석자** | 이재성, 이선호, 이지훈, 정승진, 한규영 | | |
| **불참자** | (불참의 이유도 명시할 것) | | |
| **주요안건** | 1. 회로도 작성 2. 파이썬 코드 픽스호크에서 실행 3. 비행 테스트 | | |
| **회의내용** | * **지난주 진행상황**  1. **추가 물품 주문(GPS, 자이로 센서 등)** 2. **설계 변경에 의한 메세지큐 연결** 3. **Matlab을 이용한 차량과 드론의 거리 유지 제어 블럭도 제작** 4. **비행 소프트웨어의 소프트웨어 추가 구현 및 코드 리펙토링ㅎ** 5. **GPS를 이용한 차량 속력 계산 프로그램 구현** 6. **탑제체 부착 케이스 모델링** 7. **드론 켈리브레이션 조정 완료**  * **이번주 진행상황 (진행상황을 나타내는 사진 등 첨부)**  1. **드론 바인딩 및 초도비행** 2. **픽스호크와 라즈베리파이 커넥트** 3. **모듈 전원 설계 및 제작** 4. **메세지큐 제작** 5. **Matlab을 이용한 차량-드론간의 거리 유지 제어 블럭도 보완** 6. **가속도 센서를 이용한 차량의 속도 측정**  * **지난주 피드백**  1. **제안하는 알고리즘을 시뮬레이션하는 방법에 관한 연구가 추가적으로 필요** 2. **5m 단위로 waypoint를 설정하는 것이 확보한 GPS의 성능으로 가능한지 확인해 볼 것**  * **문제점**  1. **외부전원을 통해 라즈베리파이 전원 공급시 충분한 암페어가 나오지 않음** 2. **배터리 부착부가 헐거워 비행시 x축 중심 무게중심이 이동됨** 3. **가속도 -> 속도 변환 (V = V0 + at) 과정 중에 하드웨어 오차로 인한 속도 계산 오류** 4. **커넥터 이후 코드 실행에 문제가 있음**  * **해결방안**  1. **외부전원 대신 드론의 베터리를 같이 사용하도록 제작** 2. **배터리 부착부가 고정될 수 있는 서포트 부품을 제작** 3. **Dronekit 버전이 변경되어 python 버전에 맞도록 코드 수정 필요**  * **비행 테스트**      * **전원 설계 및 제작**       **스텝 다운 모듈을 이용하여 스피커에 들어갈 12V 전류를 만듦**    **레귤레이터 모듈을 이용하여 라즈베리 파이에 들어갈 5V전류를 만듦**     * **메시지큐 구현**   **C:\Users\user\Downloads\sending.png**  **Producer 스레드에서 메시지 push**  **C:\Users\user\Downloads\full.png**  **Producer 스레드에서 큐가 가득찼을 경우 빌때까지 대기**  **C:\Users\user\Downloads\receive.png**  **Concumer 스레드에서 메시지 받아옴**  **C:\Users\user\Downloads\empty.png**  **Consumer 스레드에서 큐가 비어있으면 no data 반환**   * **픽스호크와 라즈베리파이 연결**   **픽스호크와 MAVlink를 통해 라즈베리파이와 연결함**  **터미널에서 명령어 실행확인** A screenshot of a computer  Description automatically generated <연결상태 > <연결 안될시>  * **Matlab을 이용한 차량-드론간의 거리 유지 제어 블럭도 보완**     **<Simulink 제어 블럭도>**   * Signal generator으로 펄스변화 생성 * m = 5000(kg), m\_d = 1.8(kg) 설정     **<드론과 차량의 position> <드론-차량간의 상대거리>**    **<드론의 가속도를 1.5m/으로 제한을 둔 Simulink 제어블럭도>**    드론의 가속도를 saturation을 통하여 1. 5m/로 제한하여 출력하였을 때  Position은 블럭도의 logic에 따라 로 0.5초와 1초를 집어 넣었을 시 다음과 같은 값을 얻을 수 있다.  <가속도를 saturated한 position 값>    드론의 가속도를 제한하지 않고 구할 경우 드론의 가속도는 약 10.888…m/이며, position은 블럭도의 logic에 따라 이며 다음과 같은 값이 출력 됨을 알 수 있다.  ***하지만, 드론의 최대가속도는 최대 약 2~3m/로 제한적이므로 위와 같이 saturation을 사용하여 limit 값을 정해주는 것이 옳다고 할 수 있다.***  <saturated하지 않은 position 값>    **<드론과 차량의 position> <드론-차량간의 상대거리>**  A screenshot of a computer  Description automatically generated with medium confidence  추가적인 교수님과의 미팅에서 현실적인 시뮬레이션에 대한 고민이 필요하다고 피드백을 받았다.  여러 방법 중 유니티를 이용한 시뮬레이션을 하고자 작업을 하였다.  유니티에는 기본적인 물리 엔진이 있어 힘을 이용하여 차량을 움직이고 드론을 움직일 수 있다.  또한 실제 거리를 모델링 할 수 있어 여러가지 상황의 도로를 만들 수 있다.  현재 작업은 무한한 직선의 도로와 그에 따른 웨이포인트들을 구현해 놓았고, 차량의 움직임과 물리 엔진들을 구현해두었다. 시뮬레이션 사용자가 게임처럼 키를 입력해야 차량이 움직이고 때면 서서히 멈추는 등 현실적인 물리 엔진을 구현해 두었다.   * **가속도 센서를 이용한 차량 속도 측정**   GPS 모듈은 최대 10Hz를 지원한다. 해당 프로젝트는 0.15초동안 통신이 안되면 드론은 통신 ERR라고 생각하게 되고, 이는 속도를 실시간처럼 빠르게 측정할 수 없다는 것을 의미한다. 따라서 속도를 위해서 GPS 이용을 하는 것이 아니라 가속도 센서를 활용하여 속도를 측정하기로 결정했다.  가속도 센서는 가만히 있을 때, 센서에 작용하는 지구의 중력 가속도를 X, Y, Z 축으로 벡터로 분해하여 크기를 측정하고 있고 3축의 합이 10m/s^2이 나타난다.  이론적으로 물체가 전방으로 이동한 후 멈추면 가속도는 양수에서 0으로, 그리고 음수로 바뀌며, 이에 따라 속도도 양수에서 0으로 변경된다. 그러나 앞으로 이동한 후 멈췄을 때, 센서에서 음수 값이 잘 측정되지 않는 현상이 발생했다.  음수 값이 정확하게 측정되지 않는 이유로는 센서의 보정 과정에서 오차가 발생하여 음수 값을 정확하게 측정하지 못하는 것일 수 있다. 센서의 보정 과정이 필요하다고 생각한다.  또 다른 문제로는 시간을 0.1초로 설정하고, 등가속도 운동을 가정하여 V = V0 + at 공식을 사용하여 속도를 계산하고 있다. 해당 가정이 타당한지, 또는 실제적인 적분을 사용해야 하는지에 대한 고민중에 있다.   * **다음 모임 시간/장소: 2023년 5월 21일 일** | | |